

3. Ultralyd - HC-SR04

Avstand kan måles på mange måter, ultralyd er en av dem. Her skal vi gå nærmere inn på en sensor, HC-S04. Denne sensoren virker kort sagt slik: Når den får beskjed om det, sender den ut en kort lydimpuls. Den måler hvor lang tid det tar før den får et ekko tilbake, og akkurat like lenge som det tar holder den ekkopinne høy.

Denne sensoren har, ifølge [datatabladet](#), avstandsmålinger fra 2 cm til 4 m. Den tilkobles 5 volt, jord og to signallinjer fra Arduinoen. En av signallinjene er for å fortelle den den skal starte en lydimpuls, den andre er for å måle hvor lang tid lydimpulsen tar før den kommer tilbake.

? Unknown Attachment

Hvordan sensoren egentlig ser ut

Eksempler med bibliotek

NewPing

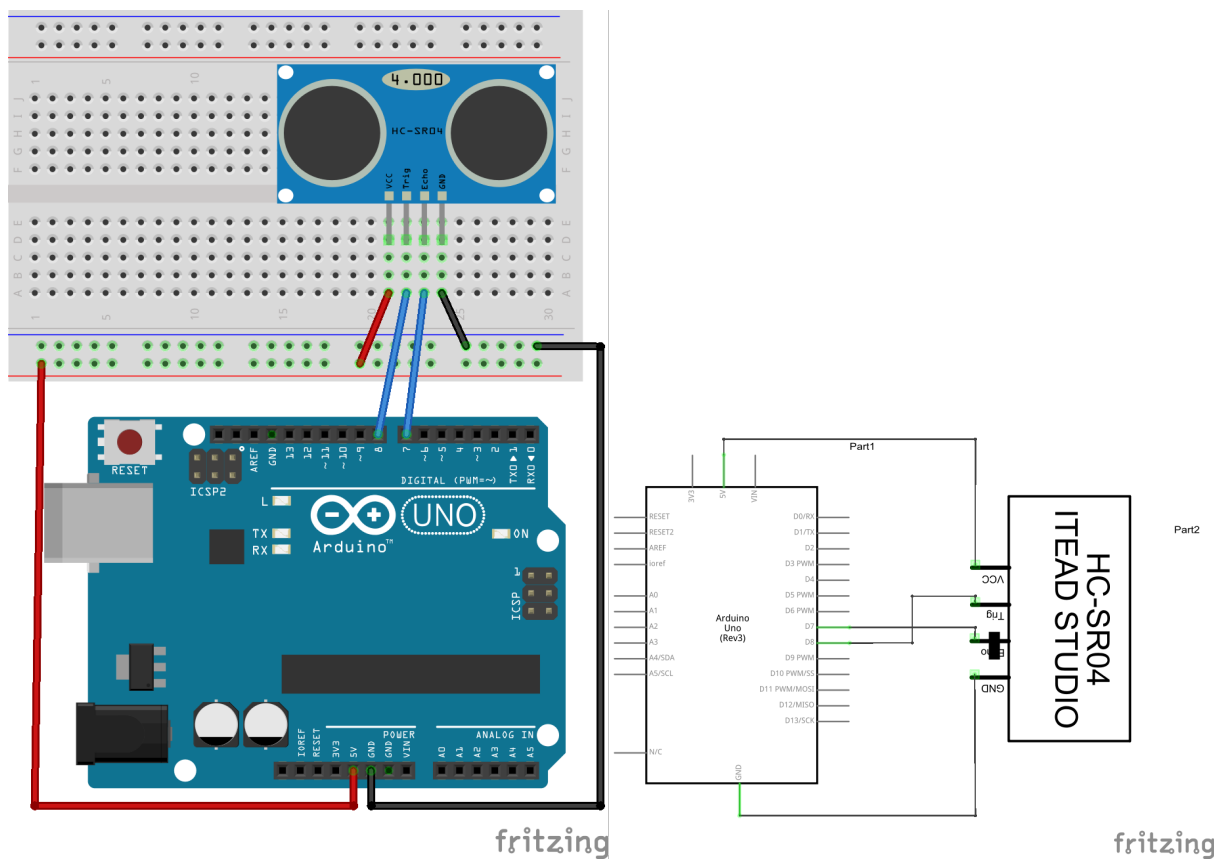
NewPing er et bibliotek som gjør det enkelt å skaffe ganske nøyaktige avlesninger. Vi har lagt det ut, og kompatibilitetsinformasjon kan finnes på informasjon om [Arduino bibliotek](#), eller på bibliotekets [egne nettsider](#).

ProximitySensor

Dette eksempelet måler avstanden jevnlig og skriver den ut.

Koden ligger i biblioteket vårt, [PLabExamples/examples/04.ProximitySensor/ProximitySensor](#).

Oppkobling gjøres slik:

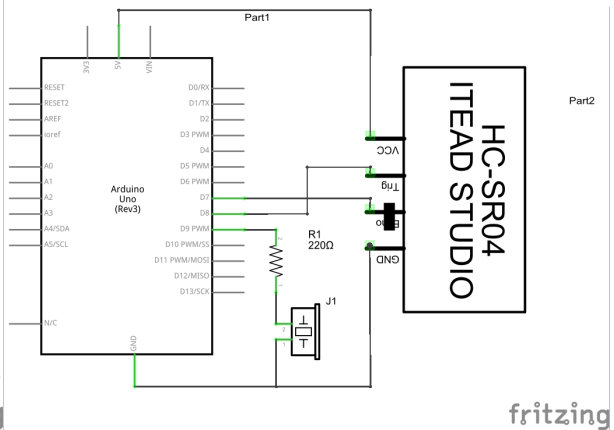
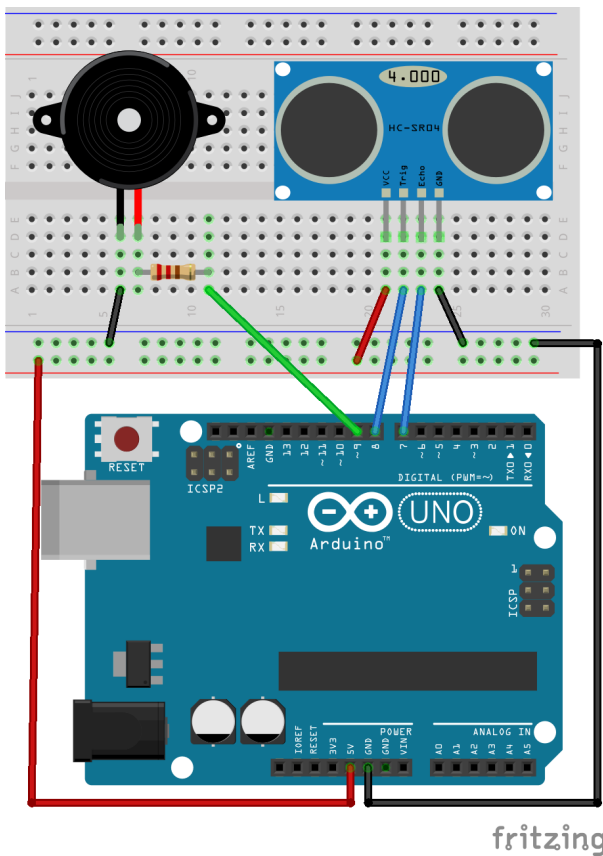


ProximitySensorBuzzer

Dette eksempelet måler avstanden jevnlig og spiller av en kort tone basert på hvor langt det målte.

Koden ligger i biblioteket vårt, [PLabExamples/examples/04.ProximitySensor/ProximitySensorBuzzer](#).

Oppkoblingen gjøres slik:

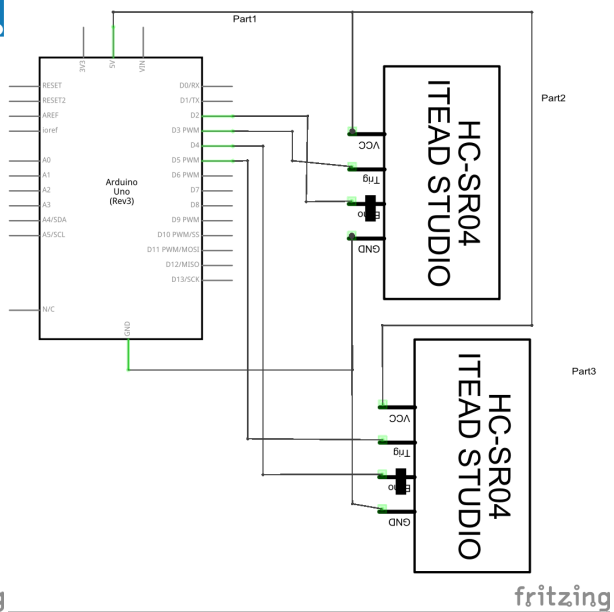
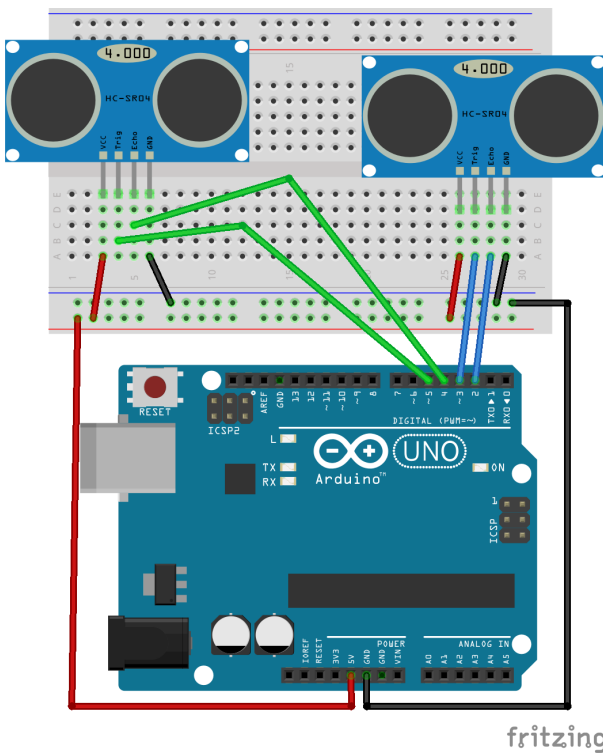


TwoSonars

Dette eksempelet måler avstanden fra to ulike ultralydsensorer hvert sekund og skriver ut resultatet til seriellvindu.

Koden ligger i biblioteket vårt, [PLabExamples/examples/04.ProximitySensor/TwoSonars](#).

Oppkoblingen gjøres slik:



Eksempel uten bruk av bibliotek

Hvis du har lyst til å teste sensoren uten å bruke bibliotek, har vi lagt med et eksempel på dette her også. Oppkoblingen er lik som ProximitySensor eksempelet.

Koden under sender som nevnt ut en kort lydimpuls før den måler hvor lang tid denne lydimpulsen bruker på å komme tilbake. I følge databladet skal vi dele tiden (som er målt i mikrosekunder) på 58 for å få avstand i cm. Ser vi på de fysiske definisjonene er dette ikke helt nøyaktig, men bra nok for dette eksempelet. Dette tallet finner vi ikke igjen i koden under, men på linje 25 har vi ett magisk tall. For å forklare: Dette eksempelet viser avstanden i meter, vi får da $(us / 58) / 100 = us * 1 / 5800$. $1/5800$ er ca. 0,00017, og vi kan derfor si $lengde = us * 0,00017$.

Ultralyd avstands eksempel uten bibliotek

```
const int echoPin = 7;
const int triggerPin = 8;

void setup() {
  Serial.begin(9600);

  // Definer hvilken type de forskjellige pinnene er
  pinMode(triggerPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  // Gjør ekkokallet
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);

  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);

  digitalWrite(triggerPin, LOW);
  // Mål hvor lang tid det tar før ekkoet kommer tilbake
  long duration = pulseIn(echoPin, HIGH);
  // Regn ut avstanden
  float distance = duration * 0.00017;

  Serial.println(distance);
  delay(1000);
}
```

Koden kan også finnes på [github kontoen](#) vår, under [Examples/UltrasoundExample](#). Her er koden mer kommentert, og utledningen av avstandsberegningen er mer detaljert.

I ekkokallet har vi to pauser vi ikke har sagt noe om så langt. Den første, på 2 mikrosekunder, er rett etter vi har garantert at signalpinnen for å starte sendingen av lydsignalet er lav. Denne forsikrer oss om at når vi sender signalet for å starte lydimpulsen så får vi et helt rent signal. Den andre pausen, på 10 mikrosekunder, er tiden vi må holde signalet høyt for at sensoren skal starte med ekkokallet. Den kan godt være lengre, men er den kortere er det ingen garanti for at lydsignalet blir sendt.

Måten vi måler tiden på er å bruke den [innebygde funksjonen pulseIn\(\)](#). Denne måler hvor lang et inngangssignal er i mikrosekunder. Denne funksjonen kan, hvis vi mister ekkoet, medføre at koden henger i ett sekund ekstra. I bruk i for eksempel roboter kan dette være et problem. Vi anbefaler derfor at dere bruker et bibliotek som hjelper til og fjerner slike forsinkelser, som biblioteket som er brukt i det andre eksempelet på denne siden.

Det bør også legges merke til at det gjøres ingen sjekk på hvor lang avstand som beregnes. Som nevnt både i datablad og i denne teksten er den gyldige avstanden mellom 2 cm og 4 m. Ved faktisk bruk bør det legges inn sikkerhetssjekker på dette, gjerne i form av enkle if..else setninger.